

Index

[Symbols](#) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

Symbols

+	addition operator
?:	bincond operator
/* */	comments - multi-line
--	comments - single-line
#	deference operator (map)
.	deference operator (tuple, bag)
::	disambiguate operator
/	division operator
==	equal operator
>	greater than operator
>=	greater than or equal to operator
<	less than operator
<=	less than or equal to operator
%	modulo operator
*	multiplication operator
!=	not equal operator
..	project-range expression
-	sign operator (negative)
+	sign operator (positive)
*	star expression
-	subtraction operator
()	type construction operator (tuple)
{ }	type construction operator (bag)

[] [type construction operator \(map\)](#)

A (top) -----

[ABS](#) function

[accumulator interface](#)

[ACOS](#) function

[AddForEach](#) optimization rule

[aggregate functions](#)

[aggregations \(hash-based\) in map task](#)

[algebraic interface](#)

aliases (for fields, relations). *See* referencing.

[Amazon S3](#)

[AND](#) (Boolean)

[arithmetic operators](#)

[ASIN](#) function

[ATAN](#) function

[autoship](#) (streaming). *See also* ship

[AVG](#) function

B (top) -----

[backward compatibility](#) (multi-query execution)

[bag functions](#)

bags (data type)

[and memory allocation](#)

[and relations](#)

[and schemas](#)

[and TOBAG function](#)

[and type construction operators](#)

[converting to string](#)

[schemas for multiple types](#)

[syntax](#)

[BagToString](#) function

[batch mode](#). *See also* memory management

[bincond operator](#) (? :)

[BinStorage](#) function

[Boolean expressions](#)

Boolean operators

[AND](#) operator

[OR](#) operator

[NOT](#) operator

[BoundScript.java](#) object

[building Pig](#)

[built in functions](#)

C ([top](#)) -----

[cache](#) (streaming)

[case sensitivity](#)

casting types

[cast operators](#)

[custom converters](#) (BinStorage)

[relations to scalars](#)

See also types tables

[CBRT](#) function

[CEIL](#) function

chararray functions (see [String Functions](#))

[checkSchema](#) method

[COGROUP](#) operator

[ColumnMapKeyPrune](#) optimization rule

[combiner](#)

[comments](#) (in Pig Scripts)

[comparison operators](#)

compression (of data)

[handling compression](#)

[compressing results of intermediate jobs](#)

[CONCAT](#) function

constants

[and data types](#)

[and nulls](#)

[convergence](#) (Python example)

[COS](#) function

[COSH](#) function

[COUNT](#) function

[COUNT_STAR](#) function

[CROSS](#) operator

D ([top](#)) -----

[-D](#) command line option

data

[combining input files](#)

[compression](#) (handling)

[compression](#) (results of intermediate jobs)

[loading](#)

[load/store functions](#) (built in functions)

[load/store functions](#) (user defined functions)

[storing final results](#)

[storing intermediate results](#) (and HDFS)

[storing intermediate results](#) (and performance)

[working with](#)

data types. *See* types

debugging

[diagnostic operators](#)

[with exec and run commands](#)

[and Pig Latin](#)

decorators. *See* Python

deference operators

[tuple or bag](#) (.)

[map](#) (#)

[DEFINE](#) (macros) operator

[DEFINE](#) (UDFs, streaming) operator

[DESCRIBE](#) operator

[DIFF](#) function

[disambiguate operator](#) (::)

[distributed file systems](#) (and Pig Scripts)

[DISTINCT](#) operator

[DISTINCT and optimization](#)

[distributed cache](#)

[downloading Pig](#)

[DUMP](#) *See also* Store vs. Dump

[dynamic invokers](#)

E ([top](#)) -----

embedded Pig

[and Groovy](#)

[invocation basics](#)

[invocation details](#) (compile, bind, run)

[and Java](#)

[and JavaScript](#)

[and PigRunner API](#)

[and PigServer Interface](#)

[and Python](#)

[EmbeddedPigStats](#) class

[error handling](#) (multi-query execution)

[eval functions](#) (built in functions)

[eval functions](#) (user defined functions). *See also* Java UDFs

[exec](#) command

executing Pig. *See* running Pig

execution modes

[local mode](#)

[mapreduce mode](#)

execution plans

[logical plan](#)

[mapreduce plan](#)

[physical plan](#)

[EXP](#) function

[EXPLAIN](#) operator

expressions

[Boolean expressions](#)

[field expressions](#)

- [general expressions](#)
- [and Pig Latin](#)
- [project-range expressions](#)
- [star expressions \(* \)](#)
- [tuple expressions](#)

F ([top](#)) -----

- [field expressions](#)

- fields

- [definition of](#)
 - [field delimiters](#)
 - [referencing](#)
 - [referencing complex types](#)

- [FILTER](#) operator

- [FILTER](#) and performance

- [filter functions](#)

- [FilterLogicExpressionSimplifier](#) optimization rule

- [flatten](#) operator

- [FLOOR](#) function

- [FOREACH](#) operator

- [fs](#) command

- [FsShell](#) commands

G ([top](#)) -----

- [general expressions](#)

- [getAllErrorMessages](#) method

- [getAllStats](#) method

- [getInputFormat](#) method

- [getNext](#) method

- [getOutputFormat](#) method

- globs

- [and BinStorage](#) function
 - [and LOAD](#) operator
 - [and REGISTER](#) statement

- [Groovy and embedded Pig](#)

[Groovy UDFs](#). *See also* UDFs

[GROUP](#) operator

[GroupByConstParallelSetter](#) optimization rule

[grunt shell](#)

H ([top](#)) -----

Hadoop

[FsShell commands](#)

[Hadoop globbing](#)

[HadoopJobHistoryLoader](#)

hadoop partitioner. *See* PARTITION BY

[Hadoop properties](#)

[versions supported](#)

[HDFS](#)

[help](#) command

I ([top](#)) -----

[identifiers](#) *See also* referencing

[ILLUSTRATE](#) operator

[IMPORT \(macros\)](#) operator

[INDEXOF](#) function

installing Pig

[builds](#)

[downloads](#)

[software requirements](#)

[interactive mode](#)

[isEmbedded](#) method

[IsEmpty](#) function

[is not null](#) operator

[is null](#) operator

J ([top](#)) -----

[JsonLoader](#)

[JsonStorage](#)

[Java and embedded Pig](#)

Java objects

[BoundScript.java](#)

[pig.java](#)

[PigProgressNotificationListener.java](#)

[PigStats.java](#)

JavaScript and embedded Pig

[JavaScript UDFs](#). *See also* UDFs

Java UDFs

[eval functions](#)

[accumulator interface](#)

[aggregate functions](#)

[algebraic interface](#)

[and distributed cache](#)

[and error handling](#)

[filter functions](#)

[and function overloading](#)

[and import lists](#)

[and Pig types](#)

[and reporting progress](#)

[and schemas](#)

[using the functions](#)

[writing the functions](#)

[load/store functions](#)

See also UDFs

[JOIN \(inner\)](#) operator

[JOIN \(outer\)](#) operator

joins

[inner joins](#)

[join optimizations](#)

[merge joins](#)

[merge-sparse joins](#)

[outer joins](#)

[replicated joins](#)

[self joins](#)

[skewed joins](#)

K ([top](#)) -----

keywords. *See* reserved keywords

[kill](#) command

L ([top](#)) -----

[LAST_INDEX_OF](#) function

[LCFIRST](#) function

[LIMIT](#) operator

[LIMIT](#) and [optimization](#)

[LimitOptimizer](#) optimization rule

[LOAD](#) operator

[LoadCaster](#) interface

[LoadFunc](#) class

[getInputFormat](#) method

[getNext](#) method

[LoadCaster](#) ubterface

[LoadMetadata](#) interface

[LoadPushDown](#) interface

[prepareToRead](#) method

[pushProjection](#) method

[relativeToAbsolutePath](#) method

[setLocation](#) method

[setUdfContextSignature](#) method

Load Functions. *See* [load/store functions](#)

[LoadMetadata](#) interface

[LoadPushDown](#) interface

[load/store functions](#)

[built in functions](#)

[user defined functions \(UDFs\)](#)

[local mode](#)

[LOG](#) function

[LOG10](#) function

[logical execution plan](#)

[LOWER](#) function

[LTRIM](#) function

M ([top](#)) -----

macros

- [defining macros](#)
- [expanding macros](#)
- [importing macros](#)

[map task and hash-based aggregation](#)

MapReduce

- [MapReduce job ids and Pig scripts](#)
- [setting the number of reduce tasks](#)

[mapreduce execution plan](#)

[mapreduce mode](#)

[MAPREDUCE](#) operator

maps (data type)

- [and schemas](#)
- [schemas for multiple types](#)
- [syntax](#)
- [and TOMAP function](#)
- [and type construction operators](#)

matches. *See* pattern matching

[math functions](#)

[MAX](#) function

[memory management](#). *See also* batch mode

[MergeFilter](#) optimization rule

[MergeForEach](#) optimization rule

[merge joins](#)

[merge-sparse joins](#)

[MIN](#) function

[modulo operator](#) (%)

[multi-query execution](#)

N ([top](#)) -----

names (for fields, relations). *See* referencing.

[nested blocks](#) (FOREACH operator)

[NOT](#) (Boolean)

[null operators](#)

nulls

- [and constants](#)
- [dropping before a join](#) (performance)
- [and JOIN operator](#)
- [and load functions](#)
- [operations that produce](#)
- [and Pig Latin](#)

O ([top](#)) -----**optimization rules**

- [AddForEach](#)
- [ColumnMapKeyPrune](#)
- [FilterLogicExpressionSimplifier](#)
- [GroupByConstParallelSetter](#)
- [LimitOptimizer](#)
- [MergeFilter](#)
- [MergeForEach](#)
- [PushDownForEachFlatten](#)
- [PushUpFilter](#)
- [SplitFilter](#)

OR (Boolean)**ORDER BY** operator[outputFunctionSchema](#) Python decorator[outputSchema](#) Python decorator**P** ([top](#)) -----[-P](#) command line option**PARALLEL**

- [and performance](#)
- [setting default_parallel](#)

[parameter substitution](#)**PARTITION BY**

- [and CROSS](#)
- [and DISTINCT](#)
- [and GROUP](#)
- [and JOIN \(inner\)](#)
- [and JOIN \(outer\)](#)

[pattern matching](#)

performance (writing efficient code)

[optimization rules for
performance enhancers](#)

See also Pig Latin

[physical execution plan](#)

[pig.cachedbag.memusage](#) property

[PigDump](#) function

[Piggy Bank](#)

Pig Latin

[automated generation of](#) (Python example)

[Pig Latin statements](#)

See also performance (writing efficient code)

Pig macros. *See* macros

[pig.java](#) object

[PigProgressNotificationListener](#) interface

[PigProgressNotificationListener.java](#) object

[PigRunner API](#). *See also* PigStats class

Pig Scripts

[and batch mode](#)

[and comments](#)

[and distributed file systems](#)

[and exec command](#)

[and MapReduce job ids](#)

[and run command](#)

[PigServer](#) interface

[PigStats](#) class

[EmbeddedPigStats](#) class

[getAllErrorMessages](#) method

[getAllStats](#) method

[isEmbedded](#) method

[SimplePigStats](#) class

See also PigRunner API

[PigStats.java](#) object

[Pig Statistics](#)

[pig.alias](#)

- [pig.command.line](#)
- [pig.hadoop.version](#)
- [pig.input.dirs](#)
- [pig.job.feature](#)
- [pig.map.output.dirs](#)
- [pig.parent.jobid](#)
- [pig.reduce.output.dirs](#)
- [pig.script](#)
- [pig.script.features](#)
- [pig.script.id](#)
- [pig.version](#)
- [PigStorage](#) function
- [Pig tutorial](#)
- Pig types. *See* data types
- [PigUnit](#)
- [positional notation](#)
- [prepareToRead](#) method
- [prepareToWrite](#) method
- projection
 - [example of](#)
 - [and performance](#)
- [project-range expressions](#)
- properties
 - [specifying Hadoop properties](#)
 - [specifying Pig properties](#)
- [PushDownForEachFlatten](#) optimization rule
- [pushProjection](#) method
- [PushUpFilter](#) optimization rule
- [putNext](#) method
- [Python and embedded Pig](#)
- [Python UDFs](#). *See also* UDFs
- Q** ([top](#)) -----
- [quit](#) (command)
- R** ([top](#)) -----

[RANDOM](#) function

referencing

[fields](#)

[fields and complex types](#)

[relations](#)

See also identifiers

[REGEX_EXTRACT](#) function

[REGEX_EXTRACT_ALL](#) function

[REGISTER](#) statement

regular expressions. *See* pattern matching

relations

[casting to scalars](#)

[and Pig Latin](#)

[referencing](#)

[relativeToAbsolutePath](#) method

[relToAbsPathForStoreLocation](#) method

[REPLACE](#) function

[replicated joins](#)

[requirements](#) (for Pig)

[reserved keywords](#)

[ROUND](#) function

[ROUND_TO](#) function

[RTRIM](#) function

[run](#) command

[running Pig](#)

[exec](#) command

[execution modes](#)

[execution order](#)

[execution plans](#)

[multi-query execution](#)

[run](#) command

S ([top](#)) -----

[SAMPLE](#) operator

[schemaFunction](#) Python decorator

schemas

- [for complex data types](#) (tuples, bags, maps)
- [and decorators](#) (Python UDFs)
- [and FOREACH](#)
- [and LOAD, STREAM](#)
- [ONSCHEMA clause](#) (UNION operator)
- [and Pig Latin](#)
- [and PigStorage](#) (function)
- [and return types](#) (JavaScript UDFs)
- [for simple data types](#) (int, long, float, double, chararray, bytearray)
- [unknown \(null\) schemas](#)

[set](#) command

[setLocation](#) method

[setStoreFuncUDFContextSignature](#) method

[setStoreLocation](#) method

[setUdfContextSignature](#) method

[sh](#) command

[shell commands](#)

[ship](#) (streaming). *See also* autoship

sign operators

[negative](#) (-)

[positive](#) (+)

[SimplePigStats](#) class

[SIN](#) function

[SINH](#) function

[SIZE](#) function

[skewed joins](#)

software requirements. *See* requirements

specialized joins

[merge joins](#)

[and performance](#)

[replicated joins](#)

[skewed joins](#)

[SPLIT](#) operator

[SplitFilter](#) optimization rule

[splits](#) (implicit, explicit)

[SQRT](#) function

[star expression](#) (*)

[statements](#) (Pig Latin)

statistics. *See* Pig statistics

[STORE](#) operator. *See also* Store vs. Dump

Store functions. *See* load/store functions

[StoreFunc](#) class

[checkSchema](#) method

[getOutputFormat](#) method

[prepareToWrite](#) method

[putNext](#) method

[relToAbsPathForStoreLocation](#) method

[setStoreFuncUDFContextSignature](#) method

[setStoreLocation](#) method

[StoreMetadata](#) interface

[StoreMetadata](#) interface

[Store vs. Dump](#)

[STREAM](#) operator

[streaming](#) (DEFINE operator)

[string functions](#)

[STRSPLIT](#) function

[SUBSTRING](#) function

[SUM](#) function

T ([top](#)) -----

[TAN](#) function

[TANH](#) function

[TextLoader](#) function

[TOBAG](#) function

[TOKENIZE](#) function

[TOMAP](#) function

[TOP](#) function

[TOTUPLE](#) function

[TRIM](#) function

[tuple expressions](#)

[tuple functions](#)

tuples (data type)

[and relations](#)

[and schemas](#)

[syntax](#)

[and TOTUPLE function](#)

[and type construction operators](#)

[type construction operators](#) (tuple, bag, map)

type conversions. *See* casting types, types tables

[types](#) (simple and complex)

[types and performance](#)

types tables

[for addition, subtraction](#)

[for equal, not equal](#)

[for matches](#)

[for multiplication, division](#)

[for negative \(negation\)](#)

[for nulls](#)

See also casting types

[tutorial](#) (for Pig)

U ([top](#)) -----

[UCFIRST](#) function

UDFs

[and function instantiation](#)

[and monitoring](#)

[passing configurations to](#)

[and performance](#) (Accumulator Interface)

[and performance](#) (Algebraic Interface)

[Piggy Bank](#) (repository)

[UDF interfaces](#)

See also Java UDFs, JavaScript UDFs, Python UDFs

[UNION](#) operator

[UPPER](#) function

user defined functions. *See* UDFs

[utility commands](#)

V ([top](#))

W ([top](#))

X ([top](#))

Y ([top](#))

Z ([top](#))